WEITEK

WTL 3132/WTL 3332 FLOATING POINT DATA PATH

PRELIMINARY DATA

July, 1986

The WTL 3132/3332 single chip floating point data paths offer a full instruction set, including multiply, multiply/accumulate, ALU and divide operations. Efficient design and architecture, combined with CMOS technology, provide 20 MFLOPS performance at very low power.

Related products: 7136 32-bit Sequencer; 7137 32-bit Integer Processor.

WEITEK
TIME: 2 PM DATE: 7/22
PRELIMINARY
TRANSMITTED BY:

Contents

A

WTL 3132 AND WTL 3332 32-BIT FLOATING POINT DATA PATH

PRELIMINARY DATA
July 1986

Features

HIGH PERFORMANCE, SINGLE CHIP 32-BIT FLOATING POINT MULTIPLIER/ACCUMULATORS WITH FOUR PORT 32 x 32-BIT REGISTER FILE AND DIVIDE LOGIC UNIT

FULL FUNCTION

Add
Subtract
Multiply
Multiply/accumulate
Divide (supported by internal architecture)
Conversion to and from two's complement integer
Absolute value

HIGH SPEED

WTL 3132-100 and WTL 3332-100 20 MFlops (100 ns clock)
WTL 3132-120 and WTL 3332-120 16 MFlops (120 ns clock)

ONE BI-DIRECTIONAL AND TWO UNI-DIRECTIONAL PORTS IN WTL 3332; SINGLE BI-DIRECTIONAL PORT IN WTL 3132.

UPGRADEABLE TO 40 MFLOPS

HIGH INTEGRATION FOR LOW CHIP COUNT IMPLEMENTATION

EASY TO PROGRAM (LOW LATENCY)

IEEE FLOATING POINT FORMAT

LOW-POWER CMOS TECHNOLOGY (MAXIMUM POWER LESS THAN ONE WATT)

STANDARD 144-PIN PIN GRID ARRAY (PGA) PACKAGE, WTL 3132; 168-PIN PGA, WTL 3332

Description

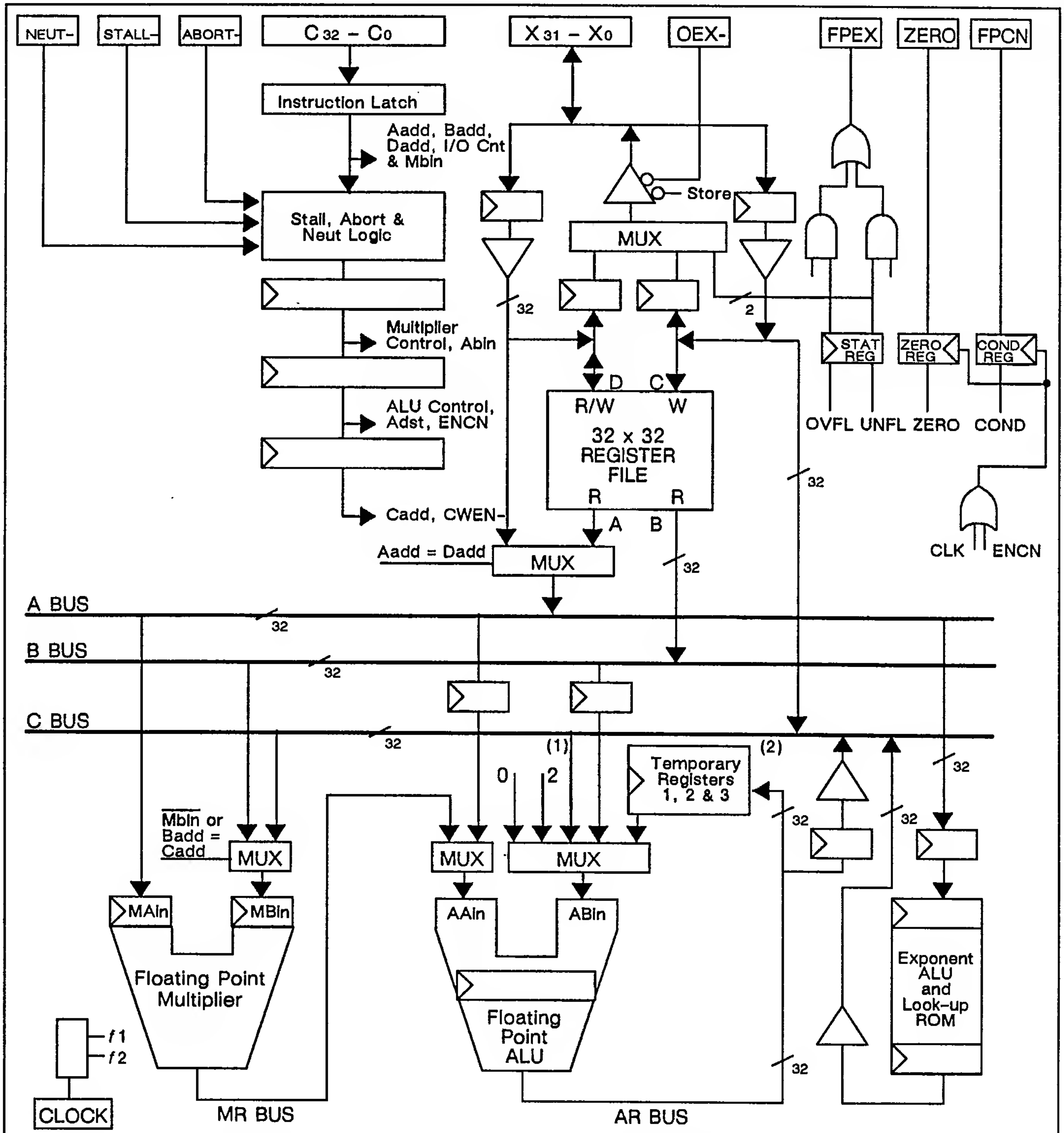
The WTL 3132 and WTL 3332 are low-power, high-speed 32-bit floating point data paths. The multiply/accumulate (MAC) function provides 20 MFlops of throughput using the IEEE Industry Standard Floating Point Format.

The devices' architecture, shown in figures 1 and 2, includes a three-port input/output (I/O) stage for the WTL 3332 and a single I/O port for the WTL 3132, a four-port 32 x 32-bit register file and three temporary registers (where operands and intermediate results can be stored), a multiplier and ALU data path and a look-up ROM to initiate Newton-Raphson division.

The WTL 3132/3332 offer a complete floating point solution for high-speed, low cost digital signal processing and graphics applications. The WTL 3132/3332-100 perform a multiplication and accumulation operation every 100 ns. Multiplication, ALU and MAC operations have an internal register-to-register latency of three clock cycles.

Implemented in low-power CMOS technology, the devices are offered in standard Pin Grid Array (PGA) packages. The WTL 3132 is packaged in a 144-pin PGA while the WTL 3332 comes in a 168-pin PGA package.

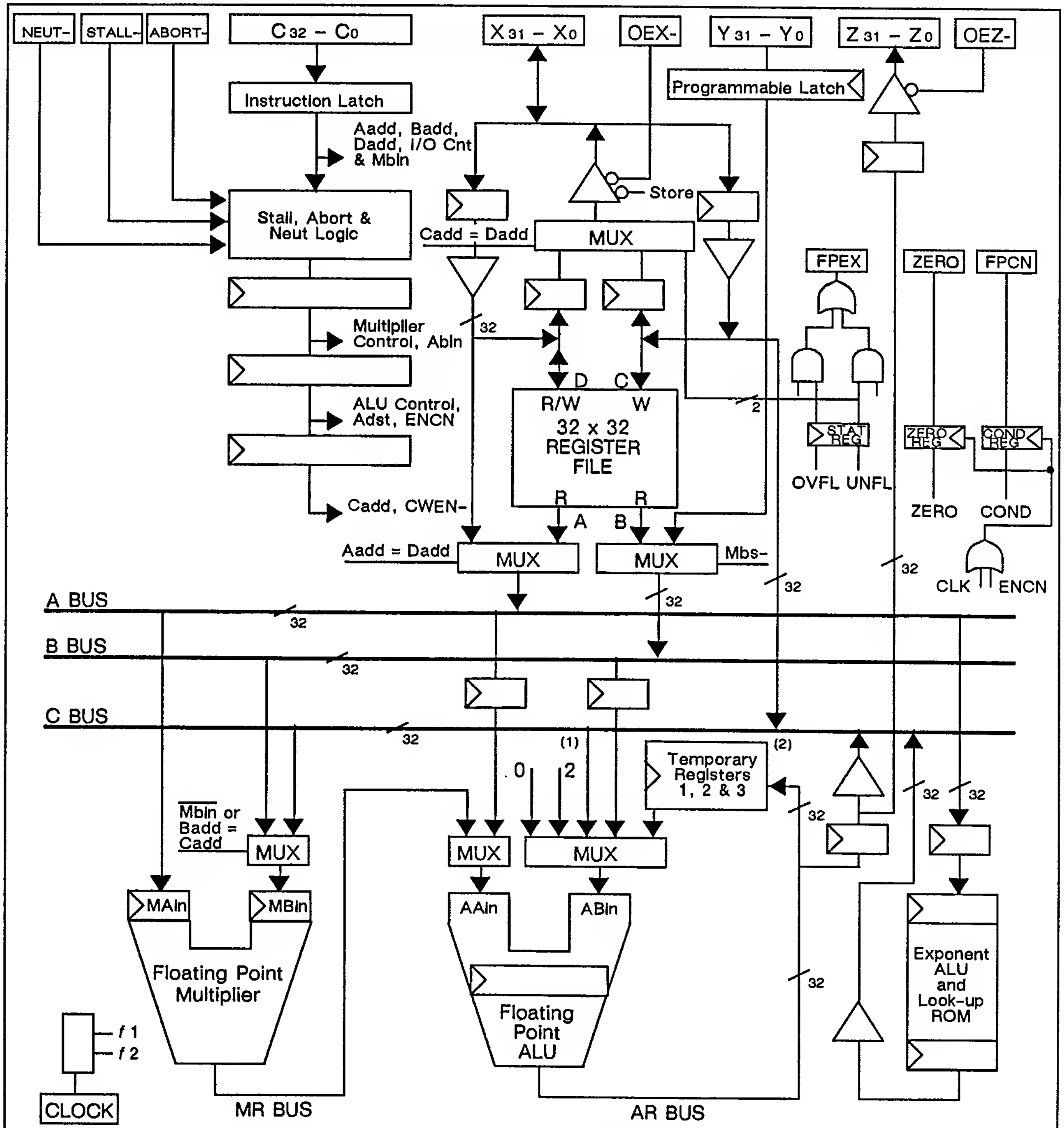
Block Diagram



NOTES:
1. The result bus (AR Bus) output register cannot be fed back into the ABin input of the ALU.
2. When a temporary register is specified as the destination for the result of an instruction, its contents will be updated two cycles later and can be used as a source to the ALU only three or more cycles later.

Figure 1. WTL 3132 Block Diagram

Block Diagram, continued



NOTES:
1. The result bus (AR Bus) output register cannot be fed back into the ABIn input of the ALU.
2. When a temporary register is specified as the destination for the result of an instruction, its contents will be updated two cycles later and can be used as a source to the ALU only three or more cycles later.

Figure 2. WTL 3332 Block Diagram

Signal Descriptions

I/O Port (X_{31-0})

32-bit input/output port

Input Port (Y_{31-0})

32-bit input port (WTL 3332 only)

Output Port (Z_{31-0})

32-bit output port (WTL 3332 only)

Code Bus (C_{32-0})/(C_{33-0})

Input to the data paths specifying the instruction to be executed. The code bus, defined by a 33-bit control word in the WTL 3132 and by a 34-bit word in the WTL 3332, is sampled on the rising edge of the clock.

X Port Output Enable (OEX–)

Active low tri-state enable control for X Port

Z Port Output Enable (OEZ–)

Active low tri-state enable control for Z Port (WTL 3332 only)

Floating Point Exception (FPEX)

Whenever an enabled exception (overflow, underflow) occurs, the Floating Point Exception output goes high.

Floating Point Condition (FPCN)

The Floating Point Condition output indicates when the result of the operation just completed by the floating point unit is less than or equal to zero.

Zero Condition (ZERO)

The Zero Condition is true when the result of an operation is exactly equal to zero.

Clock (CLK)

TTL input clock signal; all internal registers are synchronized to the rising edge of the clock.

Power (V_{DD})

Power inputs; all V_{DD} pins must be connected.

Ground (GND)

Ground connections; all GND pins must be connected.

Neutralize (NEUT–)

When Neutralize is asserted, the register file or temporary register Write, specified by the current CWEN– control, is cancelled. This signal is meant to be used in conjunction with the WTL 7136 sequencer to control delayed branching. LOAD and STORE instructions cannot be neutralized.

Stall (STALL–)

When STALL– is asserted, the Store instruction, specified by the next cycle instruction, is cancelled. This signal is intended to be used with a code cache or virtual memory to signal the delay or absence of code.

Abort (ABORT–)

When ABORT– is asserted, the register file or temporary registers WRITE instruction, specified by both the current and the next cycle instructions, are aborted. This signal is intended to be used with a data cache or virtual memory to signal the delay or absence of data. LOAD and STORE instructions cannot be aborted.

Note: The "–" notation indicates a negative true (active low) signal.

Architecture

The WTL 3132 and WTL 3332 integrate the main elements of a floating point data path onto a single chip. The three 32-bit ports in the WTL 3332 provide fast, flexible data transfers between the device and external storage. Operands and intermediate results can be stored in the internal 32 x 32-bit register file and three temporary registers. The WTL 3132/3332 include a multiplier and an ALU data path, as well as a divide logic unit. Instruction, Status, Condition and Mode registers complete the architecture.

Architecture, continued

INPUT/OUTPUT PORTS

The WTL 3132 has one 32-bit bi-directional port (X). The X Port is connected to the D Port of the register file, to the A Bus, via a multiplexer, and to the C Bus.

The WTL 3332 has the X Port plus two additional ports, one input port (Y) and one output port (Z), both 32 bits wide.

REGISTER FILE

The register file has four simultaneously accessible ports (A, B, C and D). A and B are read-only ports that can supply operands to the data path. The C Port is write only and is used to store the result of a floating point operation back into the register file or to transfer data from external memory into the register file. The D Port is a read/write port and is typically used to transfer data from external storage into the register file. Each port is controlled by a separate address field in the instruction (Aadd, Badd, Cadd and Dadd).

TEMPORARY REGISTERS

Three temporary registers are provided to accumulate intermediate results, minimizing the number of transfers to and from the register file. The temporary registers are selected by the instruction's Adst Field.

FLOATING POINT MULTIPLIER AND ARITHMETIC LOGIC UNIT (ALU)

The WTL 3132/3332 provide a high-speed, fully parallel 32-bit floating point multiplier and ALU. The multiplier inputs can be provided by any one of the three internal buses (A, B or C). These three buses, the temporary registers, the multiplier output and one of two constants (0 or 2) can provide inputs to the ALU. The register-to-register flowthrough time for a multiplication, an ALU operation or a multiply-accumulate (MAC) operation is three cycles.

DIVIDE LOGIC UNIT

The WTL 3132/3332 divide logic unit includes a look-up table in ROM that supplies an exponent and a mantissa with seven bits of precision for generating the inverse of the divisor. Starting from this initial value an accurate divisor inverse is obtained using Newton-Raphson iteration (see PROGRAMMING EXAMPLES). The division can then be implemented by multiplying the dividend by the inverse of the divisor. If the look-up table input is equal to infinity, zero is supplied as a seed; the seed will be equal to infinity if the look-up table input is equal to zero.

INSTRUCTION REGISTER

Instructions are latched into the Instruction Register on the rising edge of the clock, and are decoded and executed during the following cycles.

CONDITION AND ZERO REGISTERS

The Condition Register holds the condition of the floating point ALU operation and is available as an output on the Floating Point Condition (FPCN) pin. The Zero Register is set to one whenever the result of an operation is exactly equal to zero. It is available as an output on the Zero Pin. Both the Condition and Zero registers are affected only by those instructions whose Enable Condition Register Bit (ENCN) is asserted (set to one).

FPCN \ ZERO	0	1
0	> 0	X
1	< 0	= 0

RESULT

X: not possible

MODE REGISTER

The contents of the Mode Register configure the data path and control the two exception flags. The Mode Register is set by the Load Mode instruction (see Instruction Paragraph for details). The table on the following page describes the meaning of each mode bit.

Architecture, continued

MODE SELECTION TABLE		
MODE BIT	LOGIC VALUE	MEANING
0	0	Reserved
1	0	Reserved
2	1	Enable Underflow Flag
3	1	When Aadd = Dadd, during a Load, X (I/O) Port →A Bus
4	1	When Dadd = Cadd, during a Store, C Bus → X (I/O) Port
5	1	Enable Overflow Flag
6	0	Data and code are loaded on the same cycle (always equal to zero in the WTL 3132/3332)
7	1	Reserved
8	1	Reserved
9	0	Single load per cycle into X Port
10	1	Reserved
11	1	When Badd = Cadd, CBus → Mbin
12	0	Single Transfer Mode Selected (Y Port register latched on rising clock edge) (WTL 3332 only)

Register File Bypass

Internal logic is implemented in the WTL 3132/3332 to reduce latency and enhance performance. When data is written into the register file through one of the write ports, special internal logic allows this data to be used as an input to an operation on the same cycle. Bypassing the register file in this fashion reduces the operation time by one cycle. The WTL 3132/3332 contain three separate bypass circuits which can be independently enabled or disabled through bits in the Mode Register.

When M_3 is set to one and the A Address and D Address of a Load/Operate instruction are equal, the X Port input register is directly connected to the A Bus. Data still may be written into the register file via the D Port. This mode saves one cycle of latency in writing data from the X Port onto the A Bus.

When M_4 is set to one, the D Address of a Store/Operate instruction is equal to the C Address of the instruction specified three cycles earlier and CWEN– is low, the C Bus output register is connected to the X Input/Output Port. This saves one cycle of latency when writing data from the C Bus onto the X Port. If M_4 is set to zero, data is always read from the D Port into the X Input/Output Port during a Store/Operate instruction.

When M_{11} is set, the B Address of the current instruction is equal to the C Address of the instruction specified three cycles earlier and CWEN– is low, the register file is bypassed and the C Bus is connected directly to the B input of the multiplier. This mode of operation is useful when the output of the ALU must be fed back into the multiplier without going through the register file and adding a cycle to the total latency (as when performing a polynomial expansion).

All timing diagrams and descriptions in this document assume that the three bypass circuits are enabled.

Architecture, continued

Double Operand Transfer Mode (WTL 3332 only)

When M_9 is low only one new input per cycle can be loaded into the X Port. When M_9 is high two new inputs can be loaded into the X Port every cycle, one on the rising edge and the other on the falling edge of the clock. The data loaded on the rising edge of the clock can be written into the register file via the D Port, while the data loaded on the falling edge of the clock goes on the C Bus and cannot be written into the register file. (See Figure 17 on page 23.)

When M_{12} is set the Y Port Register latches data on the falling edge of the clock cycle. The X Port and the Y Port can then be used together to implement a double transfer per cycle: a 32-bit value is clocked into the X Port at the rising edge of the clock cycle and a second 32-bit value is loaded into the Y Port at the falling edge of the clock cycle. The result can then be written back into the register file via the C Bus.

Coprocessor Load Mode (WTL 3132A/3332A only)

M_6 must always be set to zero in the WTL 3132/3332. In the WTL 3132A/3332A, however, M_6 can be set to one thereby selecting coprocessor load mode. In this mode code is presented to the device on cycle N and the data is then loaded on the following cycle (see Figure 19 on page 24). When this mode is selected data can be loaded only from the X Port into the register file; the register file cannot be bypassed. This mode is selected whenever the WTL 3132A/3332A are used as floating point coprocessors, for example in conjunction with the WTL 7136 and WTL 7137.

STATUS REGISTER AND EXCEPTION OUTPUT

The Status Logic generates status flags for the current floating point operation. The Status Register holds and accumulates the Status Flags (overflow, underflow). instructions interrupted by the Neutralize, Abort or Stall controls don't affect the Status Register. The Status Register is read onto the bi-directional X Port by the Read Status Register function; the two least significant bits of the Input/Output Bus (X_{1-0}) specify the status.

STATUS REGISTER		
BIT NUMBER	VALUE	MEANING
0	1	overflow
1	1	underflow

The Floating Point Exception (FPEX) flags the enabled exceptions. The FPEX output value depends on M_5, M_2 and the Status Register, as shown in the following table.

M_5	M_2	FLOATING POINT EXCEPTION OUTPUT
0	0	logical zero
1	0	overflow status
0	1	underflow status
1	1	logical "or" of under/overflow status

Functional Description

INSTRUCTION EXECUTION

All the actions required to complete a given operation are specified by the instruction at the time it is loaded into the Instruction Register. The devices delay the execution of each task by the proper number of cycles. A MAC instruction, for example, specifies the multiply/accumulate operation, the addresses of the multiplicand, the multiplier and the addend and the result's final destination. During the first cycle the selected operands are loaded into the multiplier; the multiplication is executed during the second cycle and the ALU inputs are loaded; the addition is performed and the result is available on the AR Bus at the end of the third cycle. On the fourth cycle the result is written into the register file location specified by the C Address of the instruction. The C Address field of the instruction is then delayed three cycles before becoming the effective C Port address of the register file.

INSTRUCTION FORMAT

The WTL 3132 has a 33-bit instruction format. The WTL 3332 has an extra bit (Mbs–) in the instruction, as shown in Figure 3.

1	3	5	5	5	1	2	5	3	2	1	1
Mbs–	F	Aadd	Badd	Cadd	CWEN–	I/O Cnt	Dadd	Abin SEL	Adst	Mbin–	ENCN
*	2–0	4–0	4–0	4–0		1–0	4–0	2–0	1–0		
33	32 30	29 25	24 20	19 15	14	13 12	11 7	6 4	3 2	1	0

WHERE:

Mbs–:	B Bus Input Control
F:	Function Control Field
Aadd:	A Port Address
Badd:	B Port Address
Cadd:	C Port Address
CWEN–:	C Port Control Enable
I/O Cnt:	X Port Direction Control
Dadd:	D Port Address
Abin SEL:	ALU Input Multiplexer Controls
Adst:	ALU Output Destination Controls
Mbin–:	Multiplier Input Multiplexer Control
ENCN:	Condition Register Enable Control

*Only in WTL 3332

Figure 3. WTL 3132/3332 Instruction Format

Functional Description, continued

Function Select Field

The three function bits of the instruction specify the operation to be performed, as described in the following tables.

FUNCTION SELECT TABLE					
F_2	F_1	F_0	MNEMONIC	OPERATION	DESCRIPTION
1	1	1	fmac	Multiply and accumulate	MAin x MBin + ABin → ARBus
1	1	0	fmns	Multiply, negate and subtract	–MAin x MBin – ABin → ARBus
1	0	1	fmna	Multiply, negate and add	–MAin x MBin + ABin → ARBus
1	0	0		Reserved	
0	1	1	fadd	Add	AAin + ABin → ARBus
0	1	0	fsub	Subtract	AAin – ABin → ARBus
0	0	1	fsubr	Negate and add	–AAin + ABin → ARBus
0	0	0	0	Miscellaneous	(see table below)

When the F Field is equal to (0, 0, 0), the operation to be performed is specified by the Badd Field, according to the following table.

MISCELLANEOUS OPERATIONS SELECT TABLE			
$Badd_{4-0}$	MNEMONIC	OPERATION	DESCRIPTION
01000 to 11111		Reserved	
00111	flut	Look-up Table	LUT (Aadd) → CBus
00110	fix	Float-to-Fixed	Fix (Aadd) → ARBus
00101	float	Fixed-to-Float	Float (Aadd) → ARBus
00100	fabs	Absolute Value	Abs (Aadd) → ARBus
00011	lmd	Load Mode	Mode Register Loaded
00010		Reserved	
00001	Rst	Read Status Register*	Status Register → X Port
00000	Cst	Clear Status Register	0 → Status Register

*When a Read Status Register operation is selected the I/O Port Direction control must specify a Store.

Functional Description, continued

Operand Specification Fields

Port Address Fields

The 5-bit wide Aadd, Badd, Cadd and Dadd fields specify the register file port addresses. When the F Field bits are all set to logic "0", the Badd bits define the operation to be executed (see the table on the previous page).

C Port Write Enable

When the interally delayed CWEN– bit is set low, data on the C Bus is written into the register file via the C Port.

Port Control Fields

I/O Port Direction Control

The X Port direction (Read/Write) is specified by the I/O Control Field (I/O Cnt).

I/O Cnt_{1-0}	OPERATION
11	Load data from X Port to register
10	Store data from register to X Port
01	Load data from X Port to register and C Bus
00	NOP

Both 01 and 11 specify a LOAD instruction. If I/O Cnt_{1-0} is set to 01, the contents of the X Port are loaded both into the register file, via the D Port, and into the C Bus. When this LOAD operation is specified, the result bus (AR Bus) is automatically disconnected from the C Bus. If a look-up table operation had been specified three cycles earlier, though, the output of the look-up port would be connected to the C Bus and the path from the X Port to the C Bus would be automatically disabled.

If I/O Cnt_{1-0} is set to 11 the X Port contents are loaded into the D Port but not into the C Bus. This load mode is useful when a LOAD must be performed on the same cycle as a store from the result bus (AR Bus) back into the register file.

The X Port output buffer is only enabled on the cycle following a Store instruction. OEX– deasserted overrides the internal enable and tri-states the X Port.

Data may be passed directly into the multiplier or ALU without writing into the register file by specifying NOP with $M_3 = 1$ and Aadd = Dadd.

The device input/output configuration during a data transfer from or onto the X Port (Store/Load) is affected by M_3, M_4 and the register file addresses, according to the following table.

MODE	LOAD/STORE	ADDRESSES	I/O CONFIGURATION
$M_3 = 0$	LOAD*	ANY	A PORT → A BUS, X PORT → D PORT
$M_3 = 1$	LOAD*	Aadd ≠ Dadd	A PORT → A BUS, X PORT → D PORT
$M_3 = 1$	LOAD*	Aadd = Dadd	X PORT → A BUS, X PORT → D PORT
$M_4 = 0$	STORE	ANY	D PORT → X PORT, C BUS → C PORT
$M_4 = 1$	STORE	Dadd ≠ Cadd	D PORT → X PORT, C BUS → C PORT
$M_4 = 1$	STORE	Dadd = Cadd	C BUS → X PORT, C BUS → C PORT

*If two sources (C Port and D Port) attempt to simultaneously write the same register file location, the result is undefined.

B Bus Input Control (WTL 3332 Only)

The Mbs– bit of the instruction controls the B Bus input multiplexer. When Mbs– is high, the data in the Y Port Register goes into the B Bus. If Mbs– is low, the register file content, specified by the B Address, goes onto the B Bus from the B Port of the register file.

Functional Description, continued

Multiplier And ALU Input And Output Multiplexer Control Fields

Multiplier Input Multiplexer Control

The multiplier A Input (MAin) always comes from the A Bus. The B Input (MBin) can come from either the B Bus or the C Bus. The MBin– bit of the instruction controls the multiplexer at the B Input of the multiplier data path. When MBin– is high, the C Bus is connected to the B Input of the multiplier. When MBin– is low, the multiplier B Input is connected to the B Bus. The only exception occurs when M_{11} is set to one and the C and B addresses are equal; the B Input of the multiplier is then connected to the C Bus no matter what the value of MBin– is.

MBin–	SELECTED INPUT TO MBin
1	C Bus
0	B Bus

ALU Input Multiplexer Control

The ALU A Input (AAin) is connected to the A Bus when a simple ALU operation is performed. When a MAC instruction is processed, the output of the multiplier is automatically connected to AAin. The ALU B Input (ABin) can be provided by either the B Bus, the C Bus, the temporary registers or by one of two constants (0 and 2), depending on the configuration of the ALU B Input multiplexer. The ALU B Input Selection Field (ABin Sel) of the instruction configures the multiplexer according to the following table.

ABin Sel $_{2-0}$	SELECTED INPUT TO ABin
1 1 1	0
1 1 0	2
1 0 1	Reserved
1 0 0	TReg 3 (2)
0 1 1	TReg 1 (2)
0 1 0	TReg 2 (2)
0 0 1	B Bus
0 0 0	C Bus (1)

See figures 1 and 2 for appropriate notes.

There are three possible sources for the C Bus — the X Port, the look-up ROM output and the AR Bus Register. When instructions conflict, the priorities described here apply. The result of a look-up operation has the highest priority. If no look-up operation is completing, and I/O Cnt_{1-0} is set ot 01, the X Port feeds the C Bus. When neither the divide look-up table nor the X Port connection is enabled, the AR Bus directly drives the C Bus.

ALU Output Destination Control

The ALU Output Destination Control Field (Adst) determines whether the ALU output is stored in one of the temporary registers.

Adst $_{1-0}$	AR Bus DESTINATION
11	C Bus
10	TReg 1, C Bus
01	TReg 2, C Bus
00	TReg 3, C Bus

NOTE: In the WTL 3332 the ALU output is sent to the Z Output Port whenever OEZ– is set low.

Condition Enable Control

The Condition Enable control (ENCN) determines whether the Floating Point Condition and the Zero Condition, generated by the instruction, will affect the Condition and Zero registers. Only if ENCN is set high will the Condition and Zero registers be updated by the instruction's Condition Output.

Mode Initialization

The Mode Register contents configure the processor and enable the exception flags. While the Mode Register is loaded, no other operation, with the exception of I/O transfers, should be performed. Instructions in progress should be allowed to complete before a Load Mode Register instruction is executed. The instruction configuration for a Load Mode Register operation is shown in Figure 4.

1	3	5	5	5	1	2	5	3	2	1	1
X	000	MD 4–0	00011	MD 9–5	1	00	xxxxx	MODE 12–10	11	X	0
33	32 30	29 25	24 20	19 15	14	13 12	11 7	6 4	3 2	1	0

NOTE: MD = MODE
X: Don't care

Figure 4. WTL 3132/3332 Load Mode Register Instruction Format

As shown , when a Load Mode operation is executed, Aadd$_{4-0}$ specify M_{4-0}, Cadd$_{4-0}$ specify M_{9-5} and M_{12-10} are specified by ABin Sel$_{2-0}$.

Programming Examples

WTL 3132/3332 MICROWORD DESCRIPTION

Floating point source and destination operands may be identified as shown in the table below.

Floating point operations may be defined with operands as illustrated in the following table.

SOURCE AND DESTINATION OPERANDS	
.f0–31	Thirty–two 32-bit registers
.c0	Constant 0.0
.c1	Contents of the C Bus
.c2	Constant 2.0
.t1–3	Three temporary registers

OPERAND		FLOATING POINT OPERATION
flut	Areg Breg	Lookup approximation of inverse
fmac	Areg Breg Creg Dreg	Multiply and accumulate
fmns	Areg Breg Creg Dreg	Multiply, negate and subtract
fmna	Areg Breg Creg Dreg	Multiply, negate and add
fadd	Areg Creg Dreg	Add
fsub	Areg Creg Dreg	Subtract
fsubr	Areg Creg Dreg	Negate and add
fabs	Areg Dreg	Absolute value
fix	Areg Dreg	Float-to-fix conversion
float	Areg Dreg	Fix-to-float conversion
fnop	—	No instruction*

*To specify fnop, CWEN– must be set to one, Adst$_{1-0}$ must be set to 11, ENCN must be low, the Function Select Field must specify an addition (F_{2-0} must be set to 011) and the Abin Select Field must select 0 as ABin input (Abin Sel$_{2-0}$ must be set to 111).

Programming Examples, continued

The function codes take the first three bits of the instruction word and are sometimes augmented by additional bits. Operands fall into four classes and determine other opcode bits.

OPERAND CLASSES		
Areg	0-31	A or B Add
Breg	0-31 or c1	B, C or D Add, Mbin Sel
Creg	0-31, c0-2 or t1-t3	B Add, Abin Sel
Dreg	0-31, c1 or t1-t3	C Add, Adst

If the destination of a floating point operation is .0-31, CWEN- is asserted.

These additional operations may be performed in parallel with those above. The last two operations affect the I/O Cnt bits in the instruction word.

OPERAND	FP OPERATION
fld Breg	Load data
fst Breg	Store data

MICROCODE FOR A 4 x 4 TRANSFORMATION

The WTL 3332-160 and -100 have the same programming model.

The 16 matrix coefficients, a_{11-44}, are assumed to be in registers R16 to R31.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

Figure 5. WTL 3132/3332 4 x 4 Matrix Transformation

Programming Examples, continued

Cycle #	INSTR.	Areg	Breg	Creg	Dreg	I/O	COMMENT
1	fmac	$.f_0$	$.f_{16}$	.c0	$.t_1$	fld $.f_0$	; $a_{11} \cdot x$
2	fmac	$.f_0$	$.f_{17}$	.c0	$.t_2$		; $a_{21} \cdot x$
3	fmac	$.f_0$	$.f_{18}$	.c0	$.t_3$		; $a_{31} \cdot x$
4	fmac	$.f_0$	$.f_{19}$	.c0	$.t_1$		; $a_{41} \cdot x$
5	fmac	$.f_1$	$.f_{20}$	$.t_1$	$.t_2$	fld $.f_1$	; $a_{11}x + a_{12}y$
6	fmac	$.f_1$	$.f_{21}$	$.t_2$	$.t_3$		; $a_{21}x + a_{22}y$
7	fmac	$.f_1$	$.f_{22}$	$.t_3$	$.t_1$		; $a_{31}x + a_{32}y$
8	fmac	$.f_1$	$.f_{23}$	$.t_1$	$.t_2$		; $a_{41}x + a_{42}y$
9	fmac	$.f_2$	$.f_{24}$	$.t_2$	$.t_3$	fld $.f_2$	; $a_{11}.x + a_{12}y + a_{13}z$
10	fmac	$.f_2$	$.f_{25}$	$.t_3$	$.t_1$		; $a_{21}.x + a_{22}y + a_{23}z$
11	fmac	$.f_2$	$.f_{26}$	$.t_1$	$.t_2$		; $a_{31}.x + a_{32}y + a_{33}z$
12	fmac	$.f_2$	$.f_{27}$	$.t_2$	$.t_3$		; $a_{41}.x + a_{42}y + a_{43}z$
13	fmac	$.f_3$	$.f_{28}$	$.t_3$	$.f_4$	fld $.f_3$	; $x' = a_{11}x + a_{12}y + a_{13}z + a_{14}w$
14	fmac	$.f_3$	$.f_{29}$	$.t_1$	$.f_5$		; $y' = a_{21}x + a_{22}y + a_{23}z + a_{24}w$
15	fmac	$.f_3$	$.f_{30}$	$.t_2$	$.f_6$		; $z' = a_{31}x + a_{32}y + a_{33}z + a_{34}w$
16	fmac	$.f_3$	$.f_{31}$	$.t_3$	$.f_7$		; $w' = a_{41}x + a_{42}y + a_{43}z + a_{44}w$

Figure 6. WTL 3132/3332 4 x 4 Matrix Transformation Microcode

The code in Figure 6 cannot be neutralized, stalled or aborted since the values in the temporary registers would then be overwritten.

WTL 3132/3332 DIVISION CODE

The seed for 1/A is provided by the on-chip look-up ROM. It contains an exponent and a mantissa with seven bits of precision. The following sequence provides B/A to ~22 bits of precision using the Newton-Raphson algorithm.

NOTATION:

A	= divisor	→	f_5
R0	= seed for 1/A	→	f_0
R1	= first approximation	→	f_1
R2	= second approximation		
B	= dividend	→	f_6
B/A	= result	→	f_7

ALGORITHM:

First Iteration: $R1 = R0 \bullet (2 - A \bullet R0)$

Second Iteration: $R2 = R1 \bullet (2 - A \bullet R1)$

Programming Examples, continued

Cycle #	INSTR.	Areg	Breg	Creg	Dreg	I/O	COMMENT
1	flut	.f5 ,	.f0				; Get seed R0 for 1/A from look-up ROM
2	fnop						
3	fnop						
4	fmna	.f5	.f0	.c2	.f1		; 2 – A . R0
5	fnop						
6	fnop						
7	fmac	.f0	.f1	.c0	.f1		; R1 = R0 . (2 – A . R0)
8	fnop						
9	fnop						
10	fmna	.f5	.f1	.c2	.f1		; 2 – A . R1
11	fnop						
12	fnop						
13	fmac	.f1	.f1	.c0	.f1		; R2 = R1 . (2 – A . R1)
14	fnop						
15	fnop						
16	fmac	.f6	.f1	.c0	.f7		; B . 1/A
17	fnop						
18	fnop						
19	fnop					fst. f 7	; Store result
20	fnop						; Result B/A on X Port

Figure 7. WTL 3132/3332 Divide Microcode

If $M_0 = 1$, $M_4 = 1$ and $M_{11} = 1$ the divide code shown above is invalid for systems using the Stall, Abort and Neutralize features.

WTL 3132/3332 SQUARE ROOT CODE

Assuming a seed is provided by an external look-up table with an exponent and a mantissa accurate to seven bits, the following Newton-Raphson algorithm will provide a mantissa with 22 bits of precision.

NOTATION:

A	=	Operand	→	.f4
R0	=	Seed	→	.f0
R1	=	First Approximation	→	.f1
R2	=	Second Approximation	→	.f2
0.5	=	Constant	→	.f5
3.0	=	Constant	→	.f6
S	=	Results	→	.f3

ALGORITHM:

First Iteration: $R1 = (0.5 \times R0) \bullet (3.0 - A \bullet R0 \bullet R0)$

Second Iteration: $R2 = (0.5 \times R1) \bullet (3.0 - A \bullet R1 \bullet R1)$; inverse square root

$S = A \times R_2$; square root of A

Programming Examples, continued

Cycle #	INSTR.	Areg	Breg	Creg	Dreg	I/O	COMMENT
1	fnop					fld .f4	; Load A into f4
2	fmac	.f0	.c1	.c0	.c1	fld .f0	; Load R0 into f0, square R0
3	fmac	.f0	.c1	.c0	.f3	fld .f5	; Load 0.5 into f5, .5 * R0
4	fnop					fld .f6	; Load 3.0 into f6
5	fmna	.f4	.c1	.f6	.c1		; 3 – A . R0 . R0
6	fnop						;
7	fnop						;
8	fmac	.f3	.c1	.c0	.f1		; R1 = (.5 . R0) . (3 – A . R0 . R0)
9	fnop						;
10	fnop						;
11	fmac	.f4	.f1	.c0	.c1		; A . R1
12	fmac	.f5	.f1	.c0	.f3		; (.5) . R1
13	fnop						;
14	fmna	.f1	.c1	.f6	.c1		; 3 – A . R1 . R1
15	fnop						;
16	fnop						;
17	fmac	.f3	.c1	.c0	.f2		; R2 = (.5 . R1) . (3 – A. R1. R1)
18	fnop						;
19	fnop						;
20	fmac	.f4	.f2	.c0	.f3		; result S = A x R2 out
21	fnop						;
22	fnop						;
23	fnop					fst .f3	; Store result
24	fnop						; Square root of A on X Port

Figure 8. WTL 3132/3332 Square Root Microcode

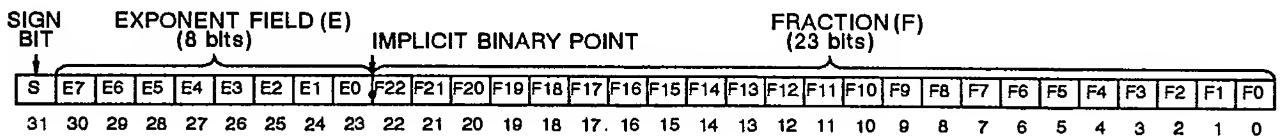
WTL 3132 AND WTL 3332 32-BIT FLOATING POINT DATA PATH

PRELIMINARY DATA
July 1986

Data Format

32-BIT FLOATING POINT (IEEE STANDARD)

The IEEE standard 32-bit floating point word is divided into three fields: a sign bit, an 8-bit exponent and a 23-bit fraction field, as shown below.



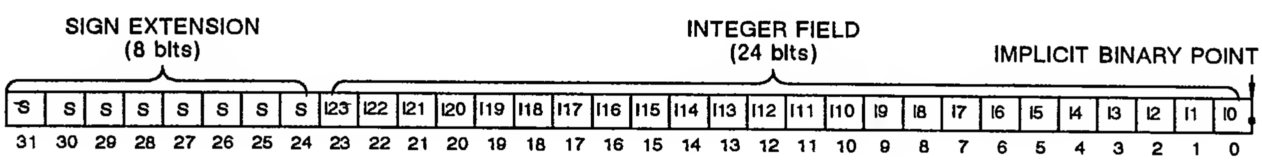
The value is determined by the following.

E	F	VALUE	NAME	MNEMONIC
255	Any	$(-1)^S$ (Infinity)	Infinity	INF
1-254	Any	$(-1)^S\ (1.F)2^{E-127}$	Normalized number	NRM
0	Any	$(-1)^S\ 0.0$	Zero	ZERO

The value contained in the 8-bit exponent field represents a power of two with a bias value of +127 added, as shown in the value chart. The fraction is multiplied by two raised to the power (−127) to produce a floating point value.

The significand field contains the 23-bit fraction and the hidden bit. Inserted during arithmetic processing, the hidden bit has a value of one for all normalized numbers and zero for zero. The fraction is the 23 bits to the right of the hidden bit. Bit F_{22} has a value of 2^{-1}; bit F_0 has a value of 2^{-23}; the hidden bit has a value of 2^0.

24-BIT FIXED POINT TWO'S COMPLEMENT



The value of the 24-bit integer field shown above can range from $2^{23} - 1$ to -2^{23} and must be sign-extended to 32 bits before performing a FLOAT operation in which the number is converted to floating point format. The eight-bit sign extension field is a repeat of bit 23, the "sign" bit of the two's complement number. When presenting two's complement integers to the WTL 3132/3332, the value of the eight sign extension bits must be consistent with the sign bit.

IEEE Considerations

The WTL 3132 and WTL 3332 comply with the IEEE Standard for Binary Floating Point Arithmetic in most respects. The differences are discussed below.

DENORMALIZED NUMBERS

Denormalized numbers have a magnitude less than 2^{-126} but greater than zero. The IEEE standard includes denormalized numbers to allow gradual underflow for operations that produce results too small to be expressed as normalized numbers. The WTL 3132/3332 do not support denormalized numbers. If the result of an operation is smaller than 2^{-126} it is replaced by zero and the Underflow Flag is asserted in the Status Register. A denormalized input is set to zero before the specified operation takes place.

NOT A NUMBER (NaN) HANDLING

The WTL 3132/3332 do not handle NaN numbers. NaN values are treated as plus or minus infinity, depending on the value of the sign bit.

OVERFLOW

An overflow occurs when an operation has a finite input operand or operands but the result is too large to be represented in the destination format. The IEEE standard requires different representation of overflows, depending on the rounding mode used. Since the WTL 3132/3332 only provides round-to-nearest mode, the WTL 3132/3332 always represent positive overflow as plus infinity and negative overflow as minus infinity. When an Overflow occurs the Overflow Flag in the Status Register is set to a logic "1".

UNDERFLOW

The IEEE standard proposes several criteria for determining if an underflow occurs. The WTL 3132/3332 generate an underflow when the result of an operation has a magnitude in the range $0 < \text{Number} < 2^{-126}$.

ROUNDING

Rounding occurs whenever an operation produces an infinitely precise result that cannot be represented exactly in the destination format. There are four rounding modes required by the IEEE standard: round to nearest, round toward plus infinity, round toward minus infinity and round toward zero.

The WTL 3132/3332 support the Round to Nearest mode: the infinitely precise result of an operation is rounded to the closest representation that fits in the destination format; if the result is exactly halfway between two representations, it is rounded to the nearest even fraction.

PRELIMINARY DATA
July 1986

Specifications

ABSOLUTE MAXIMUM RATINGS (Above Which The Useful Life May Be Impaired)

Supply voltage .... −0.5 to 7.0 V
Input voltage .... −0.5 to V_{DD}
Output voltage .... −0.5 to V_{DD}
Operating temperature range
(T_{CASE}) .... −55°C to 125°C

Storage temperature range .... −65°C to 150°C
Lead temperature (10 seconds) .... 300°C
Junction temperature .... 175°C

RECOMMENDED OPERATING CONDITIONS

PARAMETER		COMMERCIAL		MILITARY		UNIT
		MIN	MAX	MIN	MAX	
V_{DD}	Supply voltage	4.75	5.25	4.5	5.5	V
T_{CASE}	Operating temperature	0	85	−55	125	°C

DC ELECTRICAL CHARACTERISTICS, 1

PARAMETER		TEST CONDITIONS	COMMERCIAL		MILITARY		UNIT
			MIN	MAX	MIN	MAX	
V_{IHC}	High level clock input V.	V_{DD} = MAX	2.4				V
V_{ILC}	Low level clock input V.	V_{DD} = MIN		0.8			V
V_{IH}	High level input voltage	V_{DD} = MAX	2.0				V
V_{IL}	Low level input voltage	V_{DD} = MIN		0.8			V
V_{OH}	High level output voltage	V_{DD} = MIN, I_{OH} = −.4 mA	2.4				V
V_{OL}	Low level output voltage	V_{DD} = MIN, I_{OL} = 4.0 mA		0.4			V
I_{IH}	High level input current	V_{DD} = MAX, V_{IN} = V_{DD}		10			μA
I_{IL}	Low level input current	V_{DD} = MAX, V_{IN} = 0V		10			μA
I_{OZL}	Tri-state leakage current low	V_{DD} = MAX, V_{IN} = 0V		10			μA
I_{OZH}	Tri-state leakage current high	V_{DD} = MAX, V_{IN} = V_{DD}		10			μA
I_{DD}	Supply current	V_{DD} = MAX, T_{CY} = MIN TTL inputs; see Note 2		200			mA

Specifications, continued

AC SWITCHING CHARACTERISTICS, 1, 2, 3

PARAMETER		TEST CONDITIONS	31/3332-100 COMMERCIAL		31/3332-120 COMMERCIAL		31/3332-120 MILITARY		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
T_{CY}	Clock cycle time	See Figure 9	100		120		120		ns
T_{CH}	Clock high time		45		50				ns
T_{CL}	Clock low time		45		50				ns
T_S	Input setup time	See Figure 10	15		20				ns
T_{SA}	Input setup time (coprocessor mode), 4	See Figure 10	15		20				ns
T_H	Input hold time		2		2				ns
T_{DO}	Output delay time		3	30	3	35			ns
T_{ENA}	Tri-state enable time, 5	See Figure 12		30		35			ns
T_{DIS}	Tri-state disable time, 6			30		35			ns
T_{OP}	Pipelined operation time per stage		100		120		120		ns
T_{LA}	Total latency, register file to register file		300		360		360		ns

NOTES:
1. Worst case over power and temperature range.
2. TTL Input levels are 0.4 and 3.4V.
3. Timing transitions are measured at 1.5V unless otherwise noted.
4. Applies to the WTL 3132A/3332A only.
5. Device must be powered for at leasat 20 ms before testing.
6. T_{DIS} is not tested but is guaranteed by design.

Timing

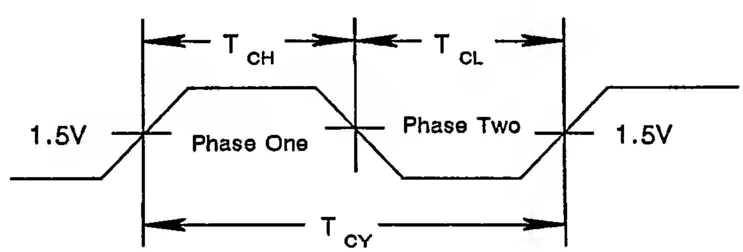


Figure 9. WTL 3132/3332 Clock Timing

Timing, continued

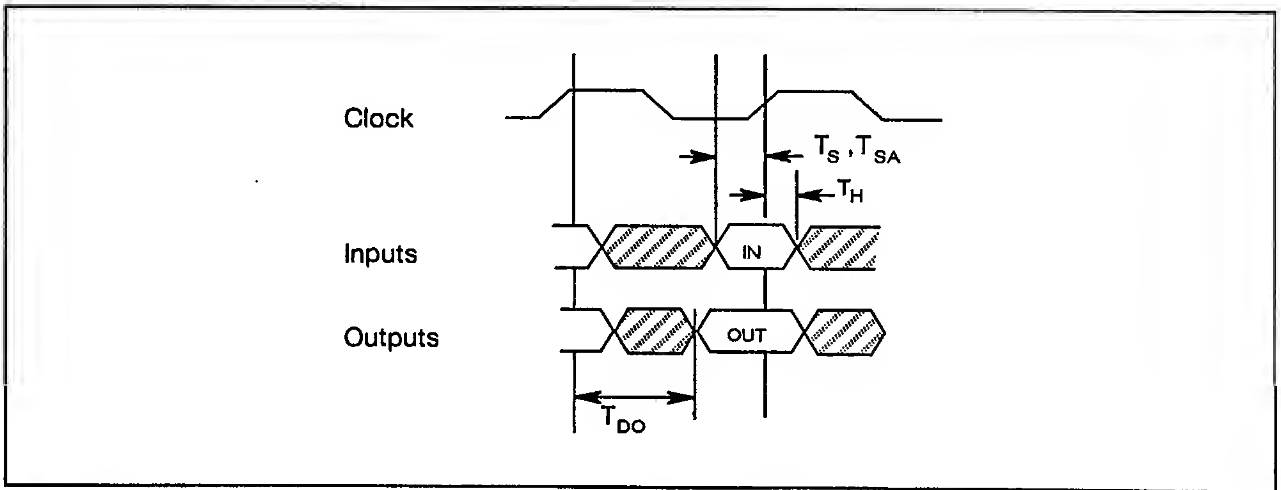


Figure 10. WTL 3132/3332 Input/Output Timing

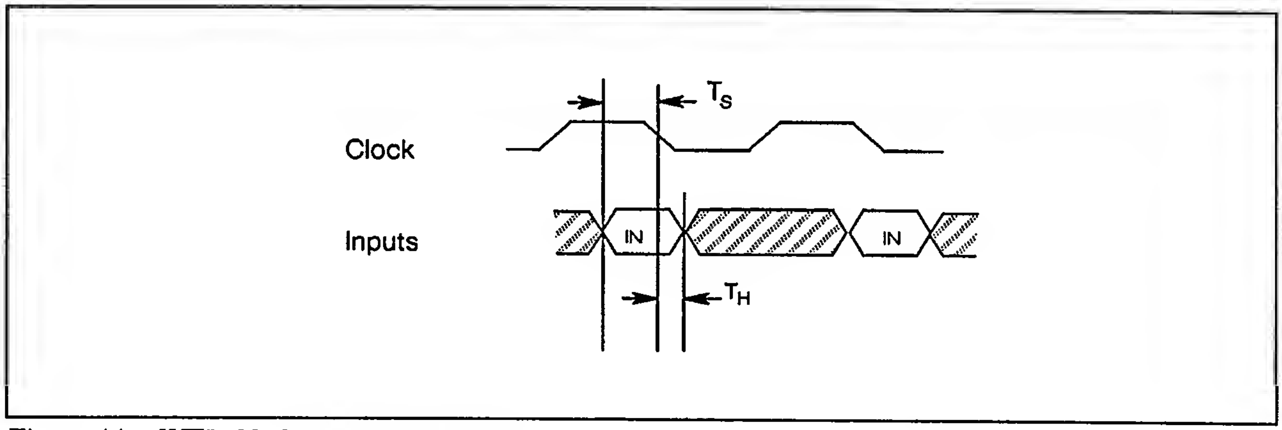


Figure 11. WTL 3332 Y Port Programmable Phase Two Latch Input Timing, $M_{12} = 1$

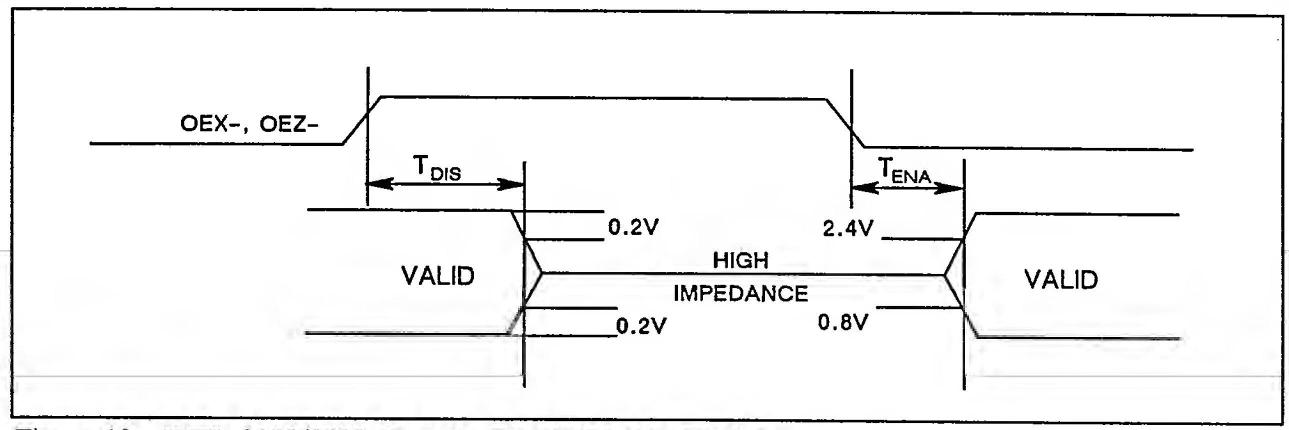


Figure 12. WTL 3132/3332 Tri-state Enable/Disable Timing

Timing, continued

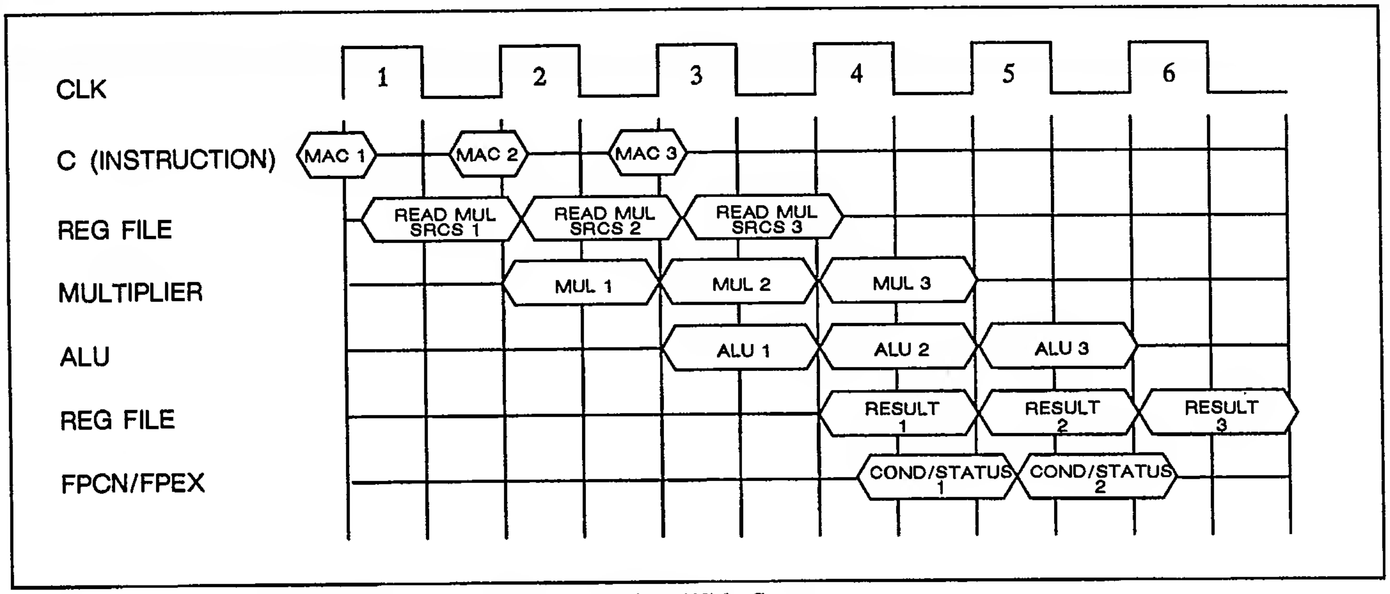


Figure 13. WTL 3132/3332 MAC Operation Timing With Status.

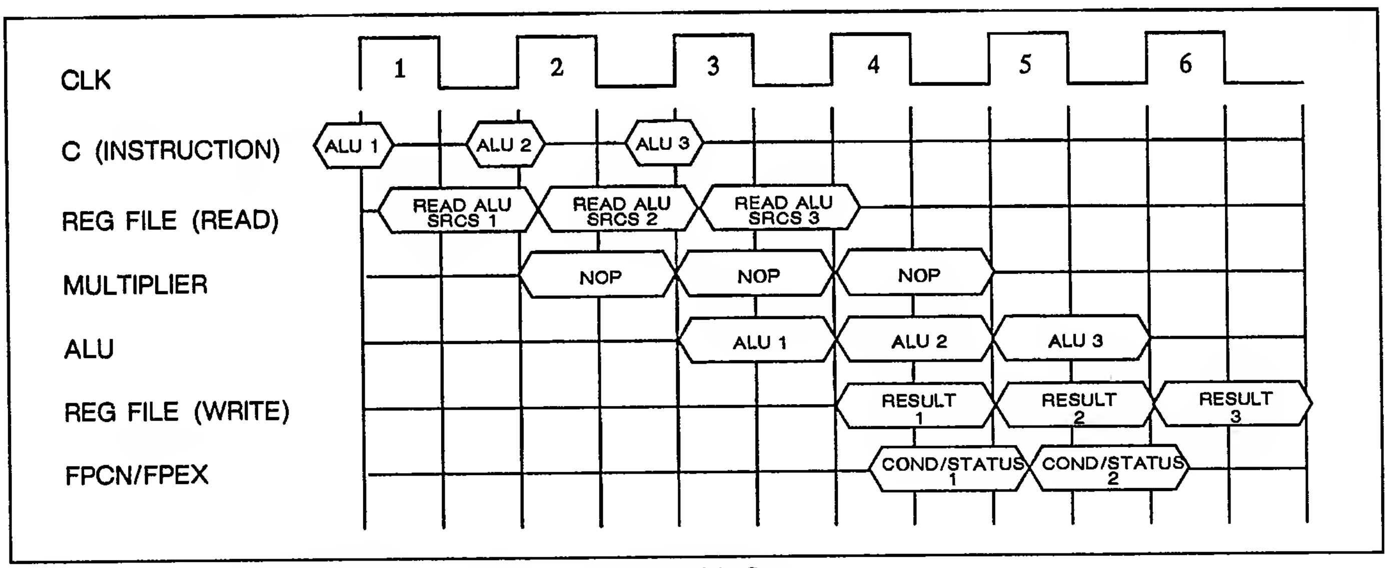


Figure 14. WTL 3132/3332 ALU Operation Timing With Status

The register-to-register latency for a MAC or ALU instruction is three cycles. Result 1 can be used on the fourth cycle as an input to the following operation due to Register Bypass Mode.

Note that even when a temporary register is specified as an operation's destination its contents cannot be used as an input to the next operation until the beginning of the fourth cycle. If it is used on the third cycle its value will be undefined.

Timing, continued

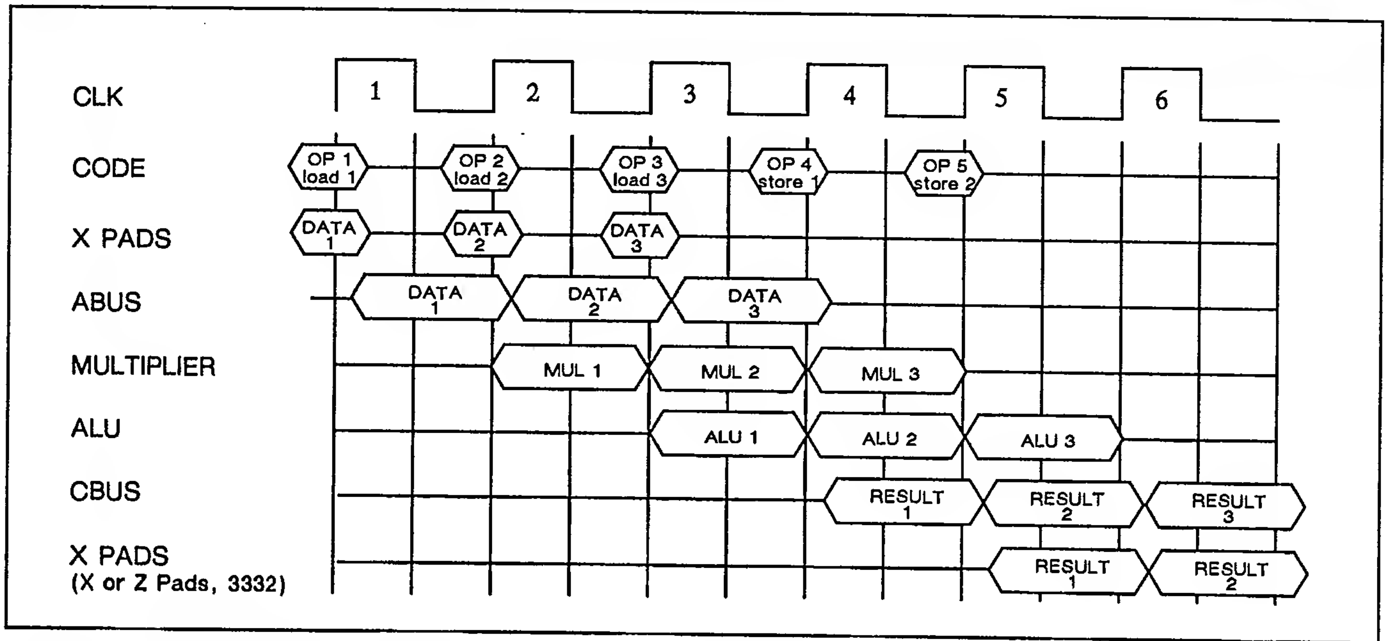


Figure 15. WTL 3132/3332 Operation Timing, External Data Path

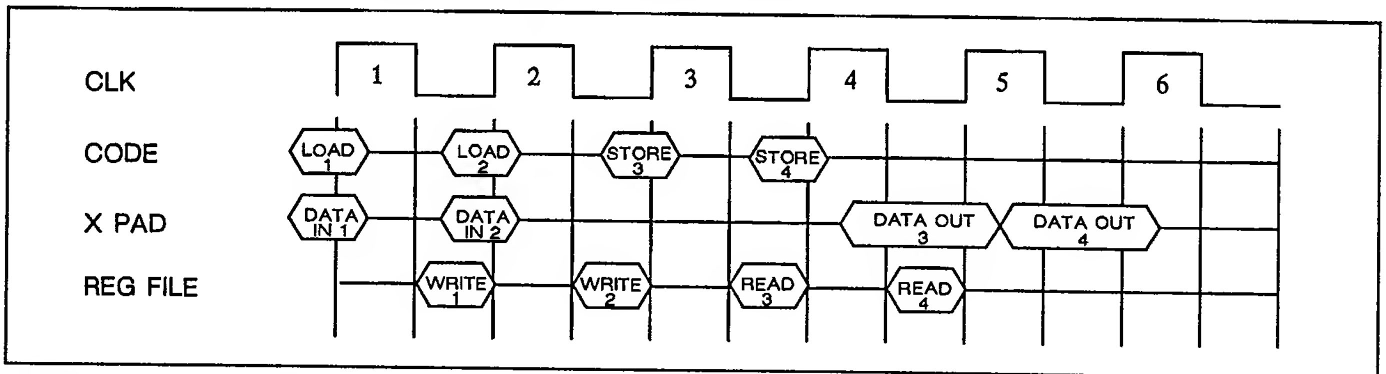


Figure 16. WTL 3132/3332 X Port Input/Output Timing, $M_9 = 0$

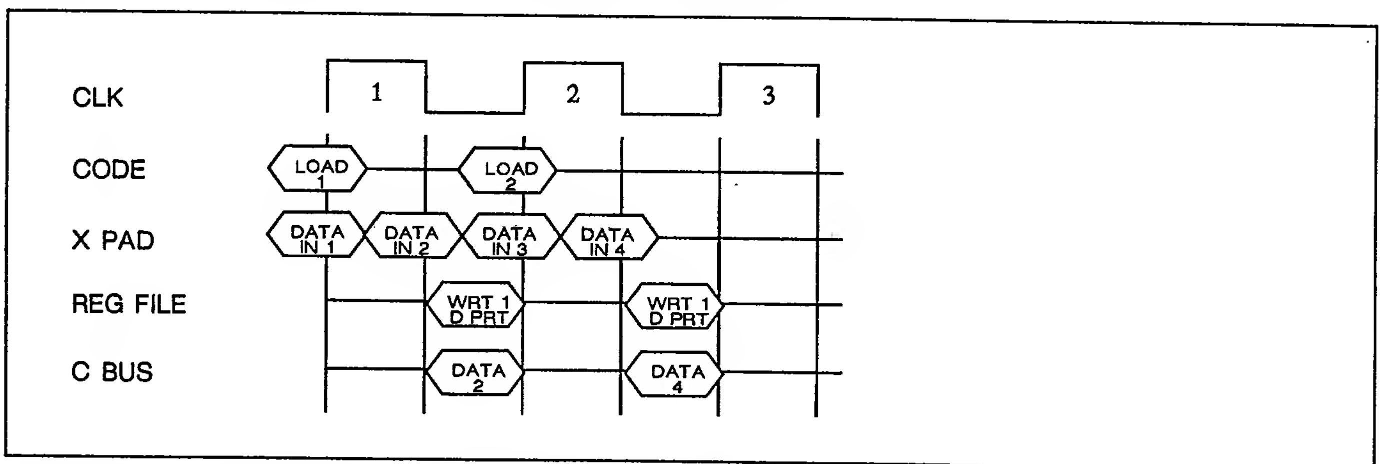


Figure 17. WTL 3132/3332 X Port Input Timing, $M_9 = 1$

Timing, continued

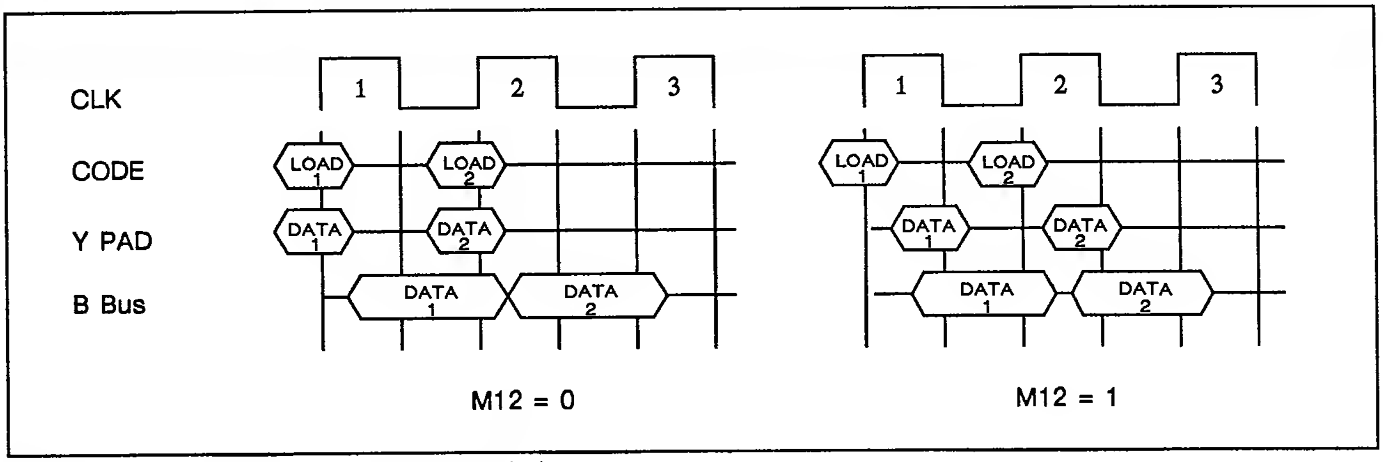


Figure 18. WTL 3332 Y Port Input Timing

The memory-to-memory latency for a MAC or ALU operation is five cycles. In Figure 15, M_4 and M_3 are assumed to be set to one and the D Address of the Store/Operate instruction and the C Address of the corresponding Load/Operate instruction are assumed to be equal. The C Bus is then connected directly to the X Port. Results are still written into the register file if the CWEN– bit of the Load/Operate instruction is set low.

From Figure 16 it is clear that a sequence of LOAD or STORE instructions or a sequence of alternated LOAD and STORE instructions can be specified without inserting NOPs. A LOAD instruction is not permitted on the second cycle following a store unless OEX– is de-asserted to prevent the WTL 3132/3332 from driving the X Port. If two or more STORE operations are specified and a LOAD must follow, for example, two NOPs must be inserted to avoid a data collision. The proper I/O control sequence is then STORE, STORE, NOP, NOP, LOAD.

The X Port input timing for $M_9 = 1$ is shown in Figure 17. Two new data inputs are loaded every cycle, one on the rising edge and one on the falling edge of the clock. Only the data loaded on the rising edge of the clock can be written into the register file; the data loaded on the falling edge is passed to the C Bus.

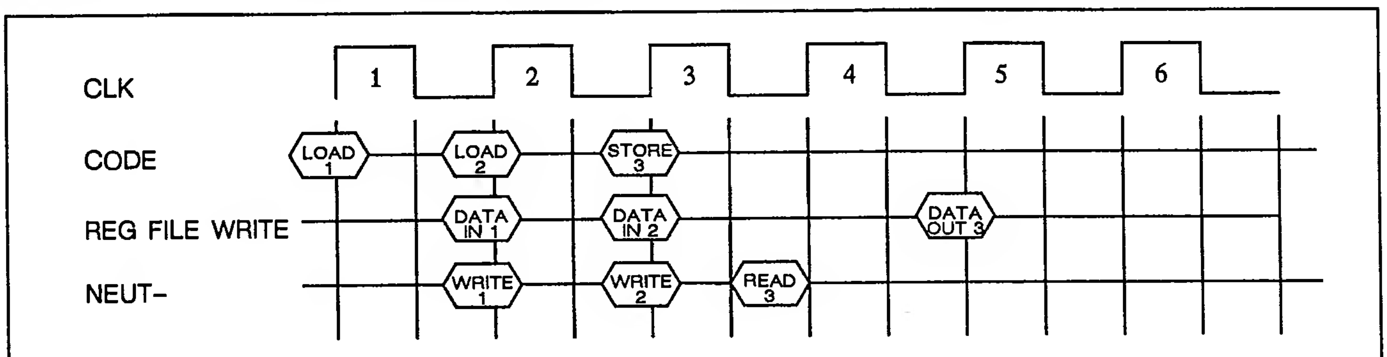


Figure 19. WTL 3132A/3332A X Port Input/Output Timing, Coprocessor Load Mode, $M_6 = 1$

In the coprocessor mode data and code are not loaded on the same cycle but on two consecutive cycles. This mode of operation is used when the WTL 3132A/3332A are used as a floating point coprocessor in conjunction with the WTL 7136 and WTL 7137.

Timing, continued

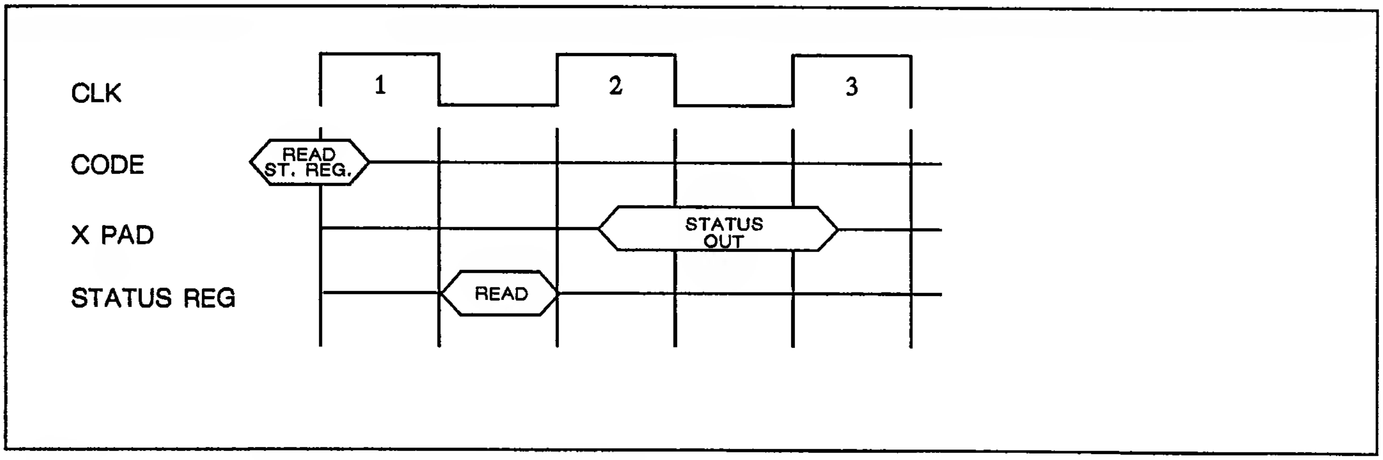


Figure 20. WTL 3132/3332 Read Status Register Timing

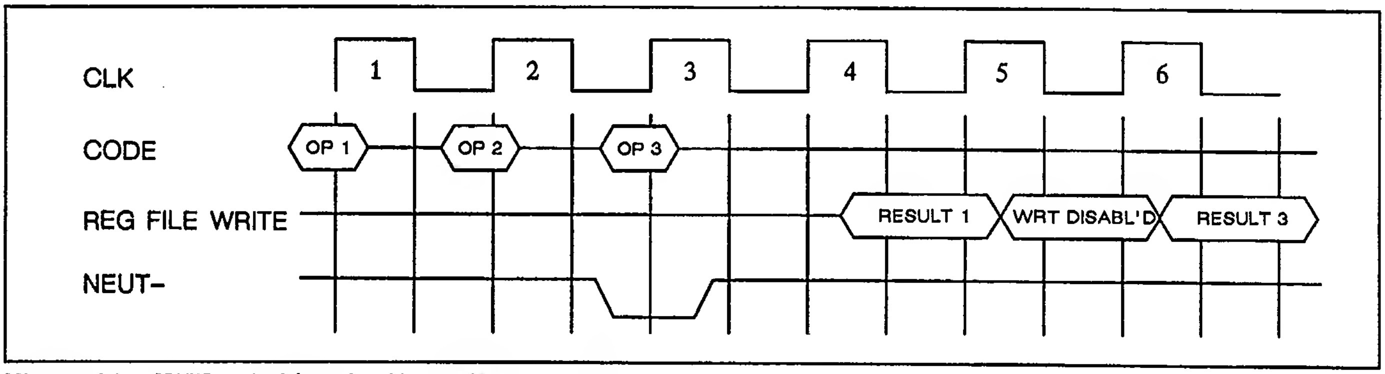


Figure 21. WTL 3132/3332 Neutralize Timing

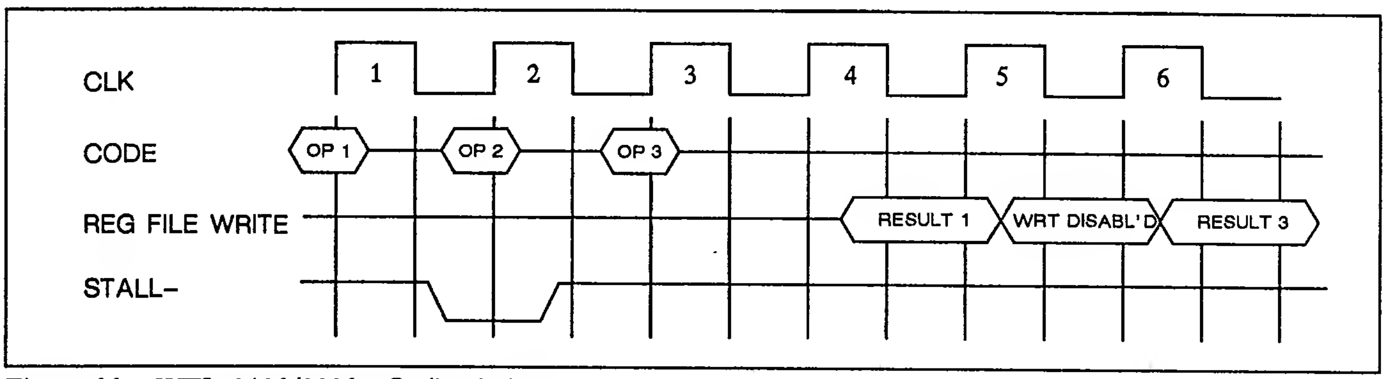


Figure 22. WTL 3132/3332 Stall Timing

Timing, continued

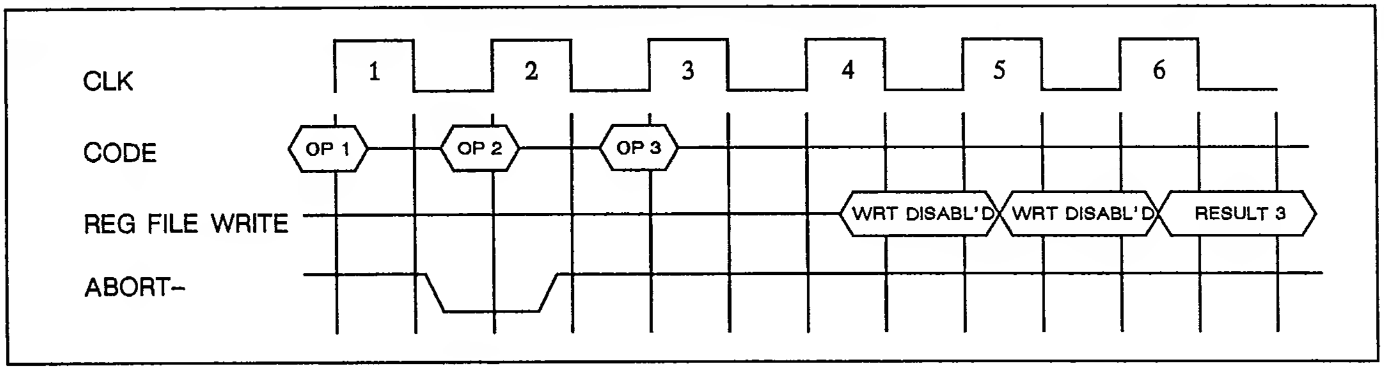


Figure 23. WTL 3132/3332 Abort Timing

STALL, NEUTRALIZE AND ABORT

Three control signals, STALL–, NEUT– and ABORT–, are used to control the execution of instructions. These signals selectively cancel combinations of the current and next instructions. They provide a simple, flexible interface to variable latency code and data memory subsystems. When an instruction is cancelled, the effects of that instruction are nullified: the C Port and temporary register are disabled; the Status register is not modified. The Condition register is still affected by the instruction. Due to the timing of I/O Cnt, LOAD and STORE operations cannot be aborted or neutralized.

The STALL– signal is used to cancel the next instruction. It is sampled at the rising clock edge and, if asserted, cancels the effects of the instruction clocked into the Instruction Register on the same clock edge. Referring to Figure 22, therefore, Instruction 2 is cancelled. This signal can be used by a code memory subsystem to stall the WTL 3132/3332 while the Code Bus is temporarily invalid.

The ABORT– signal is used to cancel the current *and* next instructions. ABORT– is sampled at the end of the cycle to which it pertains. Refer to Figure 23 for details. This signal is normally used by a data memory subsystem.

The NEUT– signal is used to cancel only the current instruction. This signal is also sampled at the end of the cycle to which it pertains. Refer to Figure 21 for details. NEUT– is normally driven by the WTL 7136 program sequencer and is used to hide the effects of delayed branching by allowing selective execution of branch shadow instructions.

For more details on the STALL–, ABORT– and NEUT– signals, refer to the WTL 7138 System Designer's Handbook.

WTL 3132 AND WTL 3332 32-BIT FLOATING POINT DATA PATH

PRELIMINARY DATA
July 1986

Pin Configuration

Pin #1 identifier	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	GND	NC	NC	X18	NC	NC	NC	X23	VDD	NC	NC	X27	NC	NC	NC
B	NC	NC	GND	X16	NC	NC	X20	X22	X24	X25	X26	NC	X29	X31	GND
C	NC	NC	X15	GND	X17	X19	X21	NC	VDD	NC	X28	NC	X30	GND	F2
D	NC	X12	X14										TIE LOW	F0	ADS T1
E	X10	X11	X13										F1	ADS T0	ABS L0
F	X8	NC	NC										ABS L2	ABS L1	ABO RT-
G	NC	X9	VDD										STA LL-	NE UT-	CW EN-
H	NC	X7	VDD					WTL 3132 TOP VIEW					CAD D2	CAD D4	CAD D3
J	X6	X5	NC										CAD D1	AAD D3	CAD D0
K	NC	NC	NC										AAD D1	AAD D2	AAD D4
L	X4	X2	NC										BAD D0	BAD D4	AAD D0
M	X3	X1	GND										DAD D2	BAD D1	BAD D3
N	NC	NC	OEX-	VDD	FPCN	GND	TIE LOW	TIE LOW	GND	GND	GND	VDD	DAD D1	DAD D4	BAD D2
P	X0	TIE LOW	GND	FPEX	ENCN	TIE LOW	clock	GND	GND	GND	GND	GND	VDD	DAD D0	DAD D3
Q	ZERO	GND	I/O CNT0	I/O CNT1	MBIn-	TIE LOW	TIE LOW	GND	GND	GND	GND	GND	GND	GND	GND

NOTE: Pins marked "Tie Low" must be connected to ground for proper WTL 3132 operation, but are reserved for use in future parts.

Pin Configuration, continued

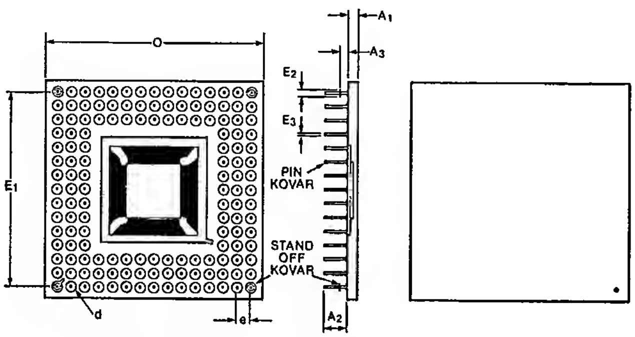
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Pin #1 identifier A	OEX-	TIE LOW	Z0	Z2	X3	Z4	NC	X5	Z7	X9	Z9	X11	Z12	X13	X15	Z15	GND
B	OEZ-	ZERO	X0	Z1	X2	NC	NC	Z5	VDD	VDD	X8	Z10	Z11	Z13	X14	GND	GND
C	GND	GND	GND	X1	Z3	X4	X6	NC	Z6	Z8	X7	X10	X12	Z14	Z16	X17	Z17
D	I/O CNT0	FPEX	FPCN												X16	Z18	X18
E	ENCN	MBIn-	I/O CNT1												Z20	X19	Z21
F	MBS-	Clock	TIE LOW												Z19	X21	X20
G	TIE LOW	TIE LOW	Y28												VDD	VDD	VDD
H	Y31	TIE LOW	Y30						WTL 3332						X23	X22	Z22
J	TIE LOW	Y29	Y26						TOP VIEW						Z23	X24	Z24
K	Y22	Y24	Y25												Z25	X25	Z26
L	Y23	Y21	Y27												VDD	VDD	VDD
M	Y18	Y19	Y20												X27	X28	Z27
N	Y16	Y15	Y17												X29	Z28	X26
P	Y14	Y13	Y12												Z31	Z30	Z29
R	Y10	Y11	Y9	DAD D3	Y6	BAD D2	AAD D4	AAD D1	CAD D0	CAD D1	Y2	Y0	STA LL-	ADS T1	F1	X31	X30
T	VDD	VDD	DAD D2	Y7	BAD D1	BAD D3	AAD D0	AAD D2	AAD D3	CAD D2	CAD D4	CW EN-	ABS L0	ABS L2	ADS T0	F0	TIE LOW
U	Y8	DAD D0	DAD D1	DAD D4	BAD D0	Y5	BAD D4	Y4	Y3	Y1	CAD D3	ABO RT-	NE UT-	ABS L1	F2	GND	GND

NOTE: Pins marked "Tie Low" must be connected to ground for proper WTL 3332 operation, but are reserved for use in future parts.

PRELIMINARY DATA
July 1986

Physical Dimensions

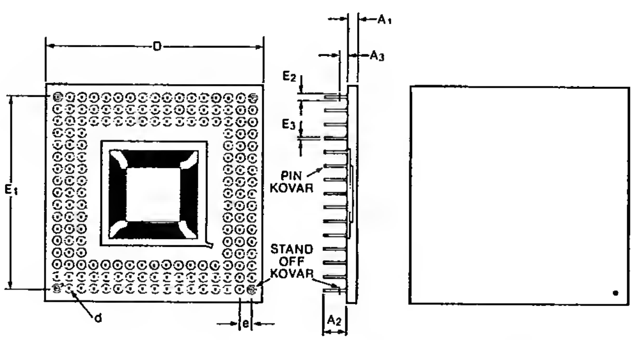
WTL 3132 144-PIN PIN GRID ARRAY



BOTTOM VIEW SIDE VIEW TOP VIEW

Symbol	DIMENSIONS	
	INCHES	MM
A1	0.080 ± 0.008	2.03 ± 0.20
A2	0.180 typ.	4.57 typ.
A3	0.050	1.27
D	1.575 sq. ± 0.016	40.0 sq. ± 0.41
E1	1.400 sq. ± 0.012	35.56 sq. ± 0.30
E2	0.050 dia. typ.	1.27 dia. typ.
E3	0.018 ±0.002	.46 ± 0.05
d	0.070 dia. typ.	1.78 dia. typ.
e	0.100 typ.	2.54 typ.

WTL 3332 168-PIN PIN GRID ARRAY



BOTTOM VIEW SIDE VIEW TOP VIEW

Symbol	DIMENSIONS	
	INCHES	MM
A1	0.095 ± 0.009	2.41 ± 0.23
A2	0.180 typ.	4.57 typ.
A3	0.050	1.27
D	1.750 sq. ± 0.018	44.5 sq. ± 0.46
E1	1.600 sq.	40.6 sq.
E2	0.050 dia. typ.	1.27 dia. typ.
E3	0.018 ±0.002	.46 ± 0.05
d	0.070 dia. typ.	1.78 dia. typ.
e	0.100 typ.	2.54 typ.

Ordering Information

PACKAGE TYPE	TEMPERATURE RANGE	ORDER NUMBER
Pin Grid Array	T_c = 0 to +85 °C	WTL 3132-120-GCD/WTL 3132-100-GCD
Pin Grid Array	T_c = 0 to +85 °C	WTL 3332-120-GCD/WTL 3332-100-GCD
Pin Grid Array	T_c = −55 to +125 °C	WTL 3132-120-GMD
Pin Grid Array	T_c = −55 to +125 °C	WTL 3332-120-GMD

Note: "A" version packs (−A00) are specified for those designs that load code on one cycle, and data on the next. Refer to the contents of this data sheet for a full explanation.

PACKAGE TYPE	TEMPERATURE RANGE	ORDER NUMBER
Pin Grid Array	T_c = 0 to +85 °C	WTL 3132–120–GCD A00/WTL 3132–100–GCD A00
Pin Grid Array	T_c = 0 to +85 °C	WTL 3332–120–GCD A00/WTL 3332–GCD–100 A00
Pin Grid Array	T_c = −55 to +125 °C	WTL 3132-120-GMD A00
Pin Grid Array	T_c = −55 to +125 °C	WTL 3332-120-GMD A00

WTL 3132 AND WTL 3332 32-BIT FLOATING POINT DATA PATH

PRELIMINARY DATA
July 1986

Revision Summary

1.	WTL 3132 Block Diagram	Revised	page 2
2.	WTL 3332 Block Diagram	Revised	page 3
3.	Condition and Zero Registers	Revised	page 5
4.	Mode Selection Table	Revised	page 6
5.	Register File Bypass	Revised	pages 6–7
6.	Double Operand Transfer Mode	Revised	page 7
7.	Function Select Table	Revised	page 9
8.	Miscellaneous Operations Select Table	Revised	page 9
9.	I/O Port Direction Control	Revised	page 10
10.	ALU Output Destination Control	Revised	page 11
11.	Mode Initialization	Revised	page 12
12.	WTL 3132/3332 Microword Description	New	pages 12–13
13.	Figure 6. WTL 3132/3332 4 x 4 Matrix Transformation Microcode	Revised	page 14
14.	WTL 3132/3332 Division Code	Revised	page 14
15.	Figure 7. WTL 3132/3332 Divide Microcode	Revised	page 15
16.	WTL 3132/3332 Square Root Code	Revised	page 15
17.	Figure 8. WTL 3132/3332 Square Root Microcode	Revised	page 16
18.	DC Electrical Characteristics	Revised	page 19
19.	AC Switching Characteristics	Revised	page 23
20.	Figure 17. WTL 3132/3332 X Port Input Timing	Revised	
21.	Figure 19. WTL 3132A/3332A X Port Input/Output Timing	New	page 24
22.	Figure 20. WTL 3132/3332 Read Status Register Timing	New	page 25
23.	WTL 3132 Pin Configuration	Revised	page 27
24.	WTL 3332 Pin Configuration	Revised	page 28
25.	Ordering Information	Revised	page 30